H

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/766,811	01/22/2001	Michele Crudele	GB920000068US1	2817

7590 08/16/2004

Manny W. Schecter
IBM Corp, Intellectual Property Law
Route 134/Kitchawan Road
Yorktown Heights, NY 10598

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 08/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/766,811	CRUDELE ET AL.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 June 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 6/01/2004.

As indicated in Applicant's response, claims 6-12 have been added. Claims 1-12 are pending in the office action.

Claim Objections

2. Claim 11 is objected to because of the following informalities: there appears to be an extraneous term in '...can are nested such that' (line 1-2). This will be interpreted as if it were '... are nested such that ...'. Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Collins, III et al., USPN: 5,845,090 (hereinafter Collins) in view of Schoening et al., USPN: 6,505,228 (hereinafter Schoening); and further in view of SmartUpdate , "SmartUpdate Developer's Guide", 1999, "http://developer.netscape.com/docs/manuals/communicator/jarman/install.htm" (hereinafter SmartUpdate); and further in view of Mishra et al., USPN: 6,389,589 (hereinafter Mishra)

As per claim 1, Collins discloses a software deployment tool (e.g. Fig. 2) cooperable with a software package including a software package file incorporating at least one action

defining respective modifications to a client processing system and at least one file required to implement such action, said tool comprising:

a plurality of objects, each object corresponding to a respective type of action (e.g. *object, methods* - col. 2, lines 33-54; col. 6, lines 9-18);

means for reading said package file and setting the attributes of objects according to the action definition in said package file (e.g. *criteria values, criteria-dependent items* -- col. 8, lines 22-39);

means for executing a check method on one of said object actions/methods to determine if a deployment can be implemented in a first mode (e.g. Fig. 8); means responsive to check failure of any object actions, for executing said actions in a second less preferable mode that the first method (e.g. UNBUILT – Fig 8; *restore* – col. 8, line 39 to col. 9, line 7 – Note: a restore instance is a uninstall mode).

But Collins does not explicitly specify that each object is a class corresponding to a respective type of action nor does Collins explicitly specify instantiating a class having attributes corresponding to the type of action of said package file; or means to execute class methods or check class failure. However, Collins discloses an object-method (col. 6, lines 9-18) association hence has suggested instantiating of class into objects with methods/attributes. Further, Collins suggest package having data and methods for installing computer program (col. 2, lines 24-42). The concept of using program to perform software installation and configuration was a known concept in the art of software distribution at the time the invention was made. Schoening, in a method to distribute software in a multi-processor network and execution of distributed components on a target machine analogous to Collins, also discloses creating of packages of

class organized in a path directory and deploying into a hierarchical scheme of class instantiated objects for executing actions during the integration of software into the target operating system or network stations (e.g. Fig. 2B,C, 2F; 3G-H; *classes contained* - Table 1, col. 11; col. 43, line 54 to col. 44, line 26; Fig. 7A). Further, analogous to Collins's method of using program for installing package components and class instantiation and action mapping by Schoening, SmartUpdate discloses instantiation of Java class objects in the Java program/script included in the installation plug-in package with *Abort, Execute, Finalize* Install methods (e.g. *Royal Airways Plug-in* - pg. 7-10). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the package installation as suggested by Collins so that it includes installation software with JAVA class instantiated object methods for checking methods failure during execution of package defined actions as suggested by Shoening and further by SmartUpdate (see SmartUpdate: *JAR* files for cross-platform – pg. 1) because object-oriented programming language used in network application and software distribution using Java-based programs are known to have helped portability across platforms and enabling resource-restraint devices to efficiently obtain software, resolve device dependencies and activate such downloaded software (see Shoening BACKGROUND of INVENTION or SmartUpdate pg. 1).

Nor does Collins explicitly disclose that the package comprises a hierarchical structure of leaf and branch nodes capable of being traversed in a top-down manner, each of such leaf nodes corresponding to the respective type of action. Collins, however discloses parsing the type of the package to allocate what type of action to take and this is similar to parsing an algorithm or condition-based structured program or script in which a specified attribute or parameter dictate

the direction to take or action to branch to (e.g. Fig. 4; col. 5, lines 1-10). Even the Jar file by SmartUpdate and its embedded scripts suggest a certain level of hierarchy among the components being packaged, such order of execution analogous to the class/action mapping top-down scheme by Schoening (e.g. Fig. 7A). The hierarchy of actions or methods to follow according to a processing/activation scheme, or according to a unpacking and installation algorithm is furthered by Mishra. Mishra, in a method to provide packages to users analogous to Collins' packaging according to some type or category, discloses package as a Class stored organized in container (e.g. Table 3, col. 7) with ids and type of class similar to Collins, and further discloses implementing this package as a hierarchy of classes (e.g. Fig. 3) with therein classes or script descriptors and state information enabling the installation and de-activation of objects installation (col. 6, line 25 to col. 11, line 30), and subsequent usage of scripts and APIs stored with the package to derive actions (e.g. Fig. 5B) similar to a tree-traversal as suggested by Collins algorithmic parsing from above. In case Collins does not provide a hierarchical structure of actions organized in a tree-like manner for enabling a parsing for actions sequences to be taken, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the hierarchy like structure Class store package with identification of binaries and state information as taught by Mishra so to enable the tree-like parsing as suggested by Collins in order to establish the actions sequence for deploying the package. The motivation for this is for determining during the hierarchical-like processing of the installation, this process as taught by Mishra can use the state of each step taken and determine appropriate action, e.g. uninstall, to be taken without incurring further action at the risk of damaging the resources of the target environment.

Nor does Collins discloses that the second method in a second mode, the 2nd mode being less preferable than the first method; but this association of mode of install with some install method as well as uninstall mode with a less desirable uninstall method has been implicitly disclosed in the teachings by Collins as well as obvious in light of the combined Collins, SmartUpdate and Mishra's teachings.

As per claim 2, Collins discloses operation to install of remove software package (e.g. col. 2, lines 24-42; UNBUILT – Fig 8; *restore* – col. 8, line 39 to col. 9, line 7).

As per claim 3, Collins in combination with Shoening does not disclose that the first mode is an undoable mode but teaches the second mode is a basic mode (see claim 2 – Note: backup and uninstall of software are the 2 basic modes of action for installing new software or upgrading software). Official notice is taken that using of non-erasable medium to store software operating small device with restraint resources (i.e. having limited or fixed amount of memory), such as firmware stored in small watch, PDA, cell phone, modem, was a known concept at the time the invention was made. In view of such concept in conjunction with the use of Java code to execute the installation and integration of cross-platform package objects into the device as suggested by Shoening or SmartUpdate, the suggestion that a form of installation not allowing any un-installation option is suggested. Hence, if the device for which the downloaded package is to be integrated happens to be a device of limited resources as mentioned above, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify Collins' method (combined with Shoening/SmartUpdate) so that one installation mode is a non-uninstall mode, i.e. undoable as claimed, because of the same reasons as recognized from the above notice.

As per claim 4, Collins discloses that the first mode is basic and the second mode is transitional (see claim 1- Note: the fact that uninstall (second mode) would be transitional is because it is temporary to a more stabilized situation wherein the software is re-installed (1st mode) after all conflicts have been resolved).

As per claim 5, this is a medium claim of claim 1, hence is rejected with the corresponding rejection as set forth therein.

5. Claims 6-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Collins, III et al., USPN: 5,845,090, in view of Mishra et al., USPN: 6,389,589.

As per claim 6, Collins discloses a system for deploying software over a network, the system comprising: a software package including at least one file, the file including definitions of actions involved in a software distribution (e.g. *scripts* – Fig. 3; col. 6, lines 10-15; *File Manifest* – col. 6, lines 46-50); a management agent configured to receive (e.g. agent *16, 17* – Fig. 2) the package and a target endpoint including a software package engine (e.g. agent *21, remote package manager 23* – Fig. 2; *Transfer Daemon* – col. 6, line 54 to col. 7, line 33) resident on the endpoint configured to receive instructions via the management agent.

But Collins does not explicitly disclose included in the package is a hierarchical structure of leaf and branch nodes capable of being traversed in a parent-child top-down manner. The tree like traversal as generating from parsing a script or an algorithmic approach as suggested by Collins and enhanced by the hierarchy of class associated with script actions and state information by Mishra has been set forth in claim 1; hence this limitation is rejected using the same rationale used therein.

As per claim 7, see Collins (col. 6, line 54 to col. 7, line 33 – Note: the act of decoding package for analyzing action based on its decoded content is inherent in the processing by Collins's software package engine).

As per claim 8, Collins does not disclose an explicit editing screen configured to define the software package, but discloses pre-build method and post-build method for administrating the package as well as including the necessary commands at build time or tool to specify component attributes (e.g. col. 6, lines 19-42; *STAGING SERVER 56* – Fig. 5A; *author, Package Tool* – col. 5, lines 2-22); hence has suggested a staging area where authoring activities by an administrator take place. Official notice is taken that a build time of any software package, the use of an editing graphical interface enabling the authoring authorities responsible for packaging the software to be built to view and modify the list of items to put together was a known concept at the time the invention was made. Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a package editing unit configured to graphically define the elements to be packaged as suggested by Collins and taught by known concepts, because this would enable more accurate process involving human-controlled or interactive, thus more selective and efficient data/components retrieval deemed appropriate by the authoring developers or package building team.

As per claim 9, Collins does not explicitly disclose serialized package, but this serializing of data being transmitted over the transmission medium was a known concept at the time the invention was made; and as such, Collins' package is considered being transmitted in serialized form has been disclosed by stream (e.g. Fig. 6).

As per claims 10 and 11, Collins does not explicitly disclose package organized in stanzas and that each stanza represents a container action. Collins teaches specifying of actions type, according to a specific package type, criteria, or profile status (e.g. Fig. 4, 7, 8), hence, the concept of organizing the main package in sub-packages, or containers, according to type (e.g. Fig. 4), or container dictating a specific group of actions has been disclosed. Besides, Mishra teaches containers (e.g. *Package container 72* – Fig. 3) and use of scripts (re claim 1) to generate actions; hence has evoked the concepts of stanzas or commands from the scripts (e.g. subroutines of Javascript or tag embedded code instructions in a markup/CGI form) representing groups of actions. Hence, for one skill in the art at the time the invention was made, in light of the organizing of package in terms of group of actions following stanzas representing package sub-classes or container from the installation scripts or installation binaries (see suggestions by Collins – col. 6, lines 46-50; col. 5, lines 1-3; see Mishra – col. 11, lines 9-30), it would have been obvious to use the teachings by using stanza in installation scripts as suggested by Mishra so to implement the use of processing of groups of actions when parsing commands by Collins in terms of script stanza representing action grouping by type, profile state, or criteria as mentioned by Collins. The motivation would be so that scripts can be modified (see Mishra: col. 11, lines 9-30) during the process of installation to efficiently accommodate for more unexpected situations and dynamically enhance resources usage in regard to the installation or backout as suggested by Collins (see Fig. 8).

As per claim 12, Collins teaches preparation site to collect separate files format or methods, and bundle methods and individual components into a package object format with manifest file (Fig. 2, 3; *STAGING SERVER 56* – Fig. 5A) and site for processing of

undifferentiated package format stream to more specific and differentiated format object for delivery to target address (e.g. Fig. 6-7); hence has disclosed preparation and test site to transform package from one format to another.

Conclusion

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label "PROPOSED" or "DRAFT" – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
August 02, 2004

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100